



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/450,755	06/01/2010	Shant H. Mardigian	PU070074	3947

24498 7590 12/13/2016

Robert D. Shedd, Patent Operations
THOMSON Licensing LLC
4 Research Way
3rd Floor
Princeton, NJ 08543

EXAMINER

CHOWDHURY, NIGAR

ART UNIT	PAPER NUMBER
----------	--------------

2484

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

12/13/2016

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

uspto@technicolor.com
pat.verlangieri@technicolor.com
russell.smith@technicolor.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte SHANT H. MARDIGIAN,
MICHAEL ZINK, and
JEREMY L. NELSON¹

Appeal 2015-007112
Application 12/450,755
Technology Center 2400

Before JASON V. MORGAN, JEREMY J. CURCURI, and
DANIEL N. FISHMAN, *Administrative Patent Judges*.

MORGAN, *Administrative Patent Judge*.

DECISION ON APPEAL

Introduction

This is an appeal under 35 U.S.C. § 134(a) from the Examiner's Final Rejection of claims 1–10 and 16–24. Claims 11–15 are withdrawn. App. Br. 17–18. We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.

Invention

Appellants disclose a user interface that allows an author of Blu-Ray Discs and other media to provide required additional information without

¹ Appellants identify THOMSON LICENSING as the real party in interest. App. Br. 3.

XML or Java coding such that authoring tools produces the relevant XML file and Java code, as needed. Abstract.

Exemplary Claim

Claim 1, reproduced below with key limitations emphasized, is illustrative:

1. A method for authoring a content-storing medium comprising:

accepting a user-entered declaration of content-related information for authoring content on the medium, the declaration of content-related information containing at least one user-created script having at least one operation;

providing a preview of user entry of the declaration of content to show progress of entry;

mapping the least one user-created script into commands to execute the at least one operation within the at least one user-created script during content payout;

generating a set of programming instructions in a programming language to execute the at least one operation within the at least one user-created script to author the content on the medium upon content payout; and

storing the set of programming instructions on the medium for execution upon playback of the content storing medium to provide information in addition to content.

Rejections

The Examiner rejects claims 1–8 and 16–23 under 35 U.S.C. § 103(a) as being unpatentable over Eklund '938 (US 2008/0238938 A1; publ. Oct. 2, 2008) and Okada (US 7,764,868 B2; iss. July 27, 2010). Final Act. 10–18, 20.

The Examiner rejects claims 9, 10², and 24 under 35 U.S.C. § 103(a) as being unpatentable over Eklund '938, Okada, and Eklund '407 (US 2008/0244407 A1; publ. Oct. 2, 2008). Final Act. 19–20.

ISSUE

Did the Examiner err in finding the combination of Eklund '938 and Okada teaches or suggests “mapping the [at] least one user-created script into commands to execute the at least one operation within the at least one user-created script during content playout” and “generating a set of programming instructions in a programming language to execute the at least one operation,” as recited in claim 1?

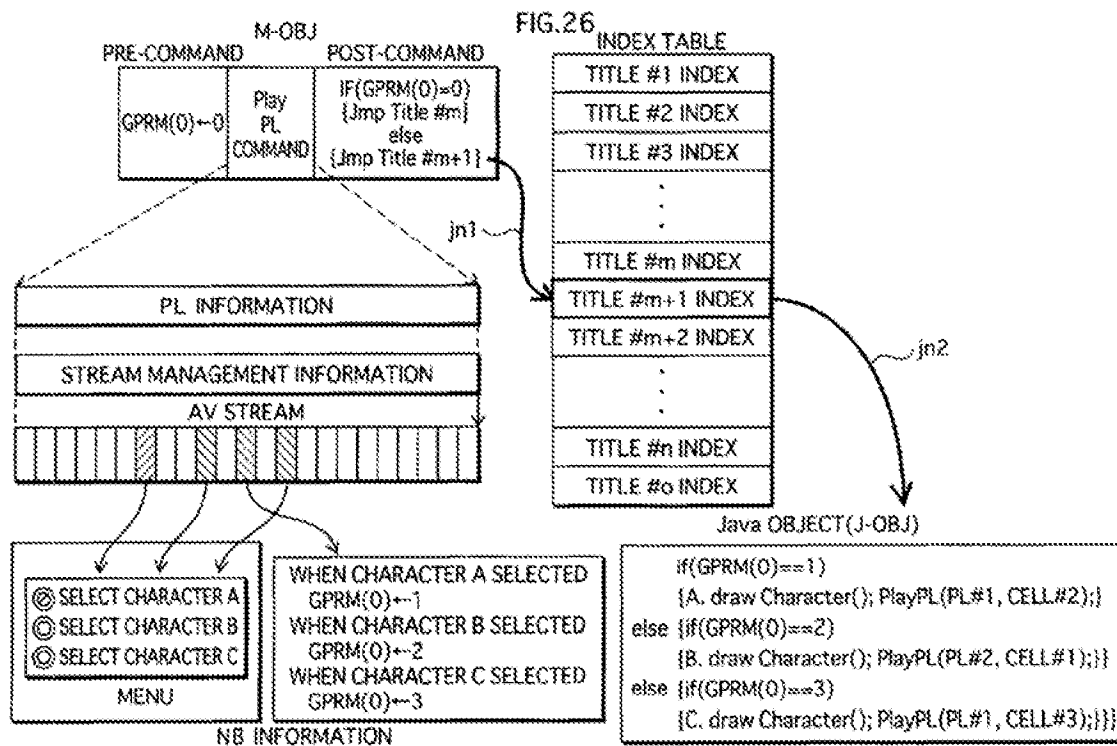
ANALYSIS

In rejecting claim 1, the Examiner finds that Okada's branching from a MOVIE object to a Java object teaches or suggests the recited *mapping at least one user-created script into commands to execute the at least one operation within the at least one user-created script during content playout*. Final Act. 13–14 (citing Okada Figs. 13–22, 26, col. 14, l. 65–col. 15, l. 19, and col. 22, ll. 45–55); Ans. 16–17. The Examiner further relies on Okada's use of Java programming instructions (i.e., the Java object) to teach or suggest *generating a set of programming instructions in a programming language to execute the at least one operation*. Final Act. 14 (citing Okada Figs. 13–22, 26); Ans. 18. The Examiner does not rely on Eklund '938 (or Eklund '407) to teach or suggest these recitations. *See, e.g.*, Final Act. 12.

² The Examiner erroneously omitted Eklund '407 from the rejection of claim 10, which depends from claim 9. We hold this error harmless.

Appellants contend the Examiner erred because Okada merely shows the use of Java instructions without describing where the disclosed program code originates. App. Br. 7. Appellants contrast this with their claimed invention, which recites “mapping of a user-created script to a set of operations, and then generating programming instructions to execute such operations.” *Id.*

Appellants’ arguments are persuasive because the Examiner relies on Okada’s use of a POST-COMMAND for instructing the playback device to perform branching to another dynamic scenario, but does not show that Okada teaches or suggests generating programming instructions to perform this operation. Final Act. 13. Rather, the Examiner merely identifies teachings and suggestions in Okada related to Java instructions in a Java OBJECT that represents a branch destination, not instructions generated to perform the branching itself. *Id.* at 13–14 (citing Okada col. 22, ll. 45–55, Figs. 13–22, 26). The distinction between the operation of branching to a new destination and the Java OBJECT that may be the destination itself is illustrated in Okada’s Figure 26, reproduced below.



Okada's Figure 26 schematically shows branching from a MOVIE object (M-OBJ) to a Java OBJECT (J-OBJ).

In addition to failing to show that the Java OBJECT described by Okada is generated to execute an operation within a user-created script (e.g., a branching operation), the Examiner does not show how the instructions depicted in the MOVIE object's POST-COMMAND are generated. Because the Examiner does not rely on Eklund '938 with respect to the disputed recitations, we are constrained by the record and agree with Appellants that the Examiner's findings do not show that the combination of Eklund '938 and Okada teaches or suggests "mapping the [at] least one user-created script into commands to execute the at least one operation within the at least one user-created script during content playout" and "generating a set of programming instructions in a programming language to execute the at least one operation," as recited in claim 1.

Accordingly, we do not sustain the Examiner's 35 U.S.C. § 103(a) rejection of claim 1, and claims 2–8 and 16–23, which are similarly rejected. Because the Examiner does not show that Eklund '407 cures the noted deficiencies of Eklund '938 and Okada, we also do not sustain the Examiner's 35 U.S.C. § 103(a) rejection of claims 9, 10, and 24.

DECISION

We reverse the Examiner's decision rejecting claims 1–10 and 16–24.

REVERSED